DYNAMIC TYPE MATING

Izak Duenyas
Matthew F. Keblis
and
Stephen M. Pollock
Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

Technical Report 94-17

June 1994
Revised October 1995

DYNAMIC TYPE MATING

IZAK DUENYAS, STEPHEN M. POLLOCK and MATTHEW F. KEBLIS

Department of Industrial and Operations Engineering

University of Michigan, Ann Arbor, MI, 48109

Abstract

We address an assembly problem, motivated by flat panel display manufacturing, where the quality (or performance) of the final product depends upon characteristics of the components to be assembled, which are not constant from component to component. We analyze the tradeoff between the increase in the potential value of products gained by putting off the "mating" of components exhibiting various characteristic "types", and the inventory costs caused by this delay in mating. We formulate this dynamic type mating problem as a Markov Decision Process and characterize the structure of the optimal policy for special cases. We then present a heuristic policy for a more general case and compare its performance against the optimal policy. Computational results indicate that the heuristic is effective for a wide variety of cases.

1 Introduction

A common feature of many manufacturing processes is the mating of two (or more) components to produce a final product. If the performance (or quality) of the final product depends upon certain characteristics not necessarily constant from component to component, then it is often desirable to put off the mating operation until a "match" of components with congruent characteristics becomes possible. Although the resulting delay in the mating of components potentially increases the value of the (eventually) mated final products, it also leads to a larger inventory of unmated components. Balancing the costs associated with carrying inventory against the value of the final products presents a classical challenge to production managers, yet the literature is surprisingly silent about this problem. In the industries of which we are aware that have this mating problem, the policies and procedures for choosing when (and how) to mate components are a matter of "rule of thumb" or, at best, explored by simulation analyses of "reasonable policies" (Iyama et al., 1992).

A typical example of the dynamic mating problem arises in flat panel display manufacturing where the two components are "active" and "passive" layers of an electronic display produced by separate machines. Because of technical constraints, it is only after two such layers are mated that the resulting electronic "sandwich" can be cut into smaller pieces to produce final products (i.e., "displays"). The final product is defective if either the active or passive layer has a defect. Since the location of defects on each layer is known after an inspection stage which occurs before the mating of the layers, each layer can be identified as being one of a set of possible "types", each containing defects in specific locations. The problem is to decide which active and passive layers (if any) to mate, given the location of known defects.

A similar problem arises in the manufacturing of ball bearings where an inner race and outer race are assembled (along with a set of balls) to produce a ball bearing (Iyama et al., 1992). Each produced race is accurately measured and classified into one of three size ranges (hence "types"). High quality bearings are then produced by mating (i.e., assembling), if possible, inner and outer races in the same size range.

Despite the importance of "type mating" in many manufacturing environments, there are very few analytical models that address this problem. The analysis of assembly queues (e.g., Ammar, 1980, Bhat, 1986, Bonomi, 1987, Duenyas 1994, Duenyas and Hopp, (1992, 1993), Duenyas and Keblis, 1995, Gershwin, 1991, Hopp and Simon 1989, Lipper and Sengupta, 1987, Lee and Pollock, 1989, Saboo and Wilhelm, 1986) generally ignores the mating problem. A common assumption in all of these analyses is that the only uncertainty in the system is about the *time* when the next component will be produced; there is no uncertainty associated with the *type* of component to be produced.

To our knowledge, the only analysis of the dynamic mating problem is by Iyama et al. (1992) who investigate the effects of an ad-hoc mating strategy (waiting for a perfect match) on buffers and machine blocking. They do not consider other mating strategies, or address generalizations beyond the content of ball bearing race mating.

Our approach is heavily influenced by our involvement with a. display manufacturer where "active" and "passive" layers of a display are to be mated. In this environment there is uncertainty with respect to the location (and number) of defects on each layer However, there is a reasonable certainty with respect to processing times and we therefore assume that the "active" and "passive" layers are produced deterministically at the same rate. (Incorporating uncertainty with respect to production time as well as "type" of each component is the subject of ongoing research).

2 Problem Formulation and Notation

We restrict ourselves to the case where two components are produced, and combined to form a final product. For convenience, we refer to these as "left" and "right" halves, respectively. The act of combining halves is called "mating". We assume that both a right half and a left half are produced simultaneously (every unit time).

When a left half is produced, it exhibits a "type" $t \in \{1, 2, ...T\}$, with probability l_t; type u right halves are produced with probability r_u, where $\sum_{t=1}^{T} l_t = 1$, and $\sum_{u=1}^{T} r_u = 1$. The probability that a right-half (left-half) is of a certain type is independent of the type of previously produced left-halves and right-halves. With these assumptions, the probability of seeing a particular "half production" output $\{t, u\}$ is $l_t r_u$. In order to measure the negative effect of holding inventory and long cycle times, we assume that each half kept in inventory for one time period has a cost h.

When a right half of type u is mated with a left half of type t, the resulting product has a value or "improvement in quality/performance" V_{tu}, where

$$V_{tu} \geq 0 \tag{2.1}$$

$$V_{tt} \geq V_{tu} \;\; for \;\; t \neq u \;\; and \;\; V_{uu} \geq V_{tu} \;\; for \;\; t \neq u \tag{2.2}$$

Inequality (2.1) implies that all matings have some value; (2.2) shows that mating left and right halves of the same type produces a maximum value. (For convenience, we define a "match" to be the mating of two halves of the same type.)

When there are three or more types, we will also assume the V_{tu} satisfy the additional inequality

$$V_{tt} + V_{uz} \geq V_{ut} + V_{tz} \quad \text{for all } z \neq u, u \neq t, z \neq t \tag{2.3}$$

This inequality ensures that immediately matching two halves of the same "type" is optimal. To see this, suppose that two halves of the same type t were held in inventory without being matched. This would imply that they would eventually be mated with some other halves (e.g., with a left half of type u and a right half of type z). However, by (2.3) this mating would result in no greater value than matching the two halves of type t and mating the two other halves of types u and z. Since there is a cost associated with holding inventory, it would be better to match the two halves of type t as soon as possible rather than hold them.

Although inequalities (2.1) through (2.3) rule out some possible value matrices, they are reasonable for a wide variety of situations including the problem that motivated this study. In flat panel display manufacturing, the highest value is obtained when layers that have defects at the same locations are mated. This is because once mated, a location is defective if it has defects on *either* layer and the display corresponding to that location will have to be discarded. For example, suppose each layer is to be divided into N=4 pieces. The left ('active') layer can be represented by a vector $\underline{x}$ of four ones or zeros denoting whether each location is defective or not. For example $\underline{x} = (1,1,0,1)$, represents a layer with a defect only in the third location. The unique type number $t(\underline{x})$ can then be represented as $t(\underline{x}) = \sum_{i=1}^{N} 2^i x_i$. Conversely, given type number t, the vector $\underline{x}(t)$ is uniquely defined to be a binary representation of t. We can similarly represent the right layer by a vector $\underline{y}$, with type $u(\underline{y})$. The value of mating/matching a left half of type t with a right half of type u is then given by $V_{tu} = r_1(\underline{x}(t) \cdot \underline{y}(u)) + r_2(N - (\underline{x}(t) \cdot \underline{y}(u)))$, where r_1 is the revenue associated with a good display, r_2 is the salvage value of a defective display and $(\underline{x}(t) \cdot \underline{y}(u))$ is the inner-product of vectors $\underline{x}(t)$ and $\underline{y}(u)$. It is straightforward to check that when $r_1 \geq r_2$this value function satisfies all three inequalities.

Other practical situations satisfying all three inequalities include those where the value of a mated product is equal to the value of the least valuable part in the product or those situations where parts with closer tolerances result in better fit and higher value (e.g., $V_{tu} = c|t - u|$), where c is a constant).

Since an optimal policy immediately matches left-halves and the right-halves of the same type, if the number of left-halves of a certain type is non-zero, then the number of right-halves must be zero. Therefore, for each type, we only need to keep track of the difference between the number of left-halves and the number of right-halves. This allows a representation of the system as being in some state: a $T - 1$ vector, $\{n_1, \ldots, n_{T-1}\}$, where n_t is the difference between the number of left halves of type t and right halves of type t. Since arrivals of left and right-halves occur at the same time, $\sum_{t=1}^{T} n_t = 0$ and therefore n_T can always be computed from the values of n_1 through n_{T-1}. The fundamental problem is to determine for any given state vector $(n_1, n_2, \ldots, n_{T-1})$, which pair (if any) of types should be mated in order to maximize net value per unit time.

3 Optimal Policy for Two Types

In this section, we characterize the optimal policy for $T = 2$ (i.e., both left and right sides exhibit only two types). The results derived in this section are used to develop a heuristic for the case with T types in Section 5.

The problem of maximizing long-run average profits (equivalently, minimizing average negative profits) is formulated as a Markov-Decision Process. From the argument above, the state of the system can be represented by a $T - 1 = 1$ vector, i.e., a scalar. Let

$g =$ the minimum long-run cost per period

$i =$ the state of the system (the difference between the number of left-halves of type 1 and right-halves of type 1).

$f_i =$ the relative cost of being in state i and following the optimal policy.

To write the MDP formulation, we need to consider the case where $i > 1$. The underlying recursive equation is:

$$g + f_i = \min \begin{cases} 2h|i| + l_1 r_1(-V_{11} + f_i) + l_2 r_2(-V_{22} + f_i) + l_1 r_2 f_{i+1} + l_2 r_1(-V_{11} - V_{22} + f_{i-1}); \\ -V_{12} + 2h|i - 1| + l_1 r_1(-V_{11} + f_{i-1}) + l_2 r_2(-V_{22} + f_{i-1}) + l_1 r_2 f_i + \\ \qquad\qquad\qquad l_2 r_1(-V_{11} - V_{22} + f_{i-2}) \end{cases} \tag{3.4}$$

In (3.4), the choice is between waiting for production of another left and right-half before making a mating decision (the upper line on the r.h.s), or mating one left half of type 1 and one right half of type 2 (the lower line). In both cases:

a) When new parts arrive, if they are of the same type, they are matched and the inventory level remains unchanged;

b) If the left-half is type 1 and the right-half type 2 then the state becomes $i + 1$;

c) If the left-half is type 2 and the right-half is type 1, then two matches are possible (left 1 and right 1, left 2 and right 2) and the state becomes $i - 1$.

Letting

$$w(i) \equiv l_1 r_1 f_i + l_2 r_2 f_i + l_1 r_2 f_{i+1} + l_2 r_1 f_{i-1} \tag{3.5}$$

$$P^+(i) \equiv 2h|i| - l_1 r_1 V_{11} - l_2 r_2 V_{22} - l_2 r_1(V_{11} + V_{22}) \tag{3.6}$$

and

$$P^-(i) \equiv 2h|i| - l_1 r_1 V_{11} - l_2 r_2 V_{22} - l_1 r_2(V_{11} + V_{22}) \tag{3.7}$$

allows (3.4) to be rewritten as

$$g + f_i = P^+(i) + \min\{w(i); -V_{12} - 2h + w(i - 1)\} \quad \text{for } i > 1 \tag{3.8}$$

Similarly, the equations for the cases where $i < -1$, $i = 1$, $i = -1$ and $i = 0$ are:

$$g + f_i = P^-(i) + \min\{w(i); -V_{21} - 2h + w(i+1)\} \quad \text{for } i < -1 \tag{3.9}$$

$$g + f_1 = P^+(1) + \min\{w(1); -V_{12} - 2h + w(0) + l_2 r_1 (V_{11} + V_{22})\} \tag{3.10}$$

$$g + f_{-1} = P^-(-1) + \min\{w(-1); -V_{21} - 2h + w(0) + l_1 r_2 (V_{11} + V_{22})\} \tag{3.11}$$

$$g + f_0 = P^+(0) + w(0) + l_2 r_1 (V_{11} + V_{22}) = P^-(0) + w(0) + l_1 r_2 (V_{11} + V_{22}) \tag{3.12}$$

Since the f_i are the relative costs of being in state i, we can arbitrarily set $f_0 = 0$.

Equations (3.8) through (3.12) completely define the MDP formulation for the 2-type problem. The structural form of the optimal policy is described by:

Theorem 1 *The optimal policy for the 2-type mating problem is a control-limit policy requiring only the state i and two numbers $i_1 \geq 0$ and $i_2 \leq 0$. That is, the optimal policy is to:*

a) mate a left half of type 1 with a right-half of type 2 if $i > i_1$;

b) to mate a left-half of type 2 with a right-half of type 1 if $i < i_2$;

c) wait for the next produced halves if $i_2 \leq i \leq i_1$.

Proof: The proof is given in Appendix A.

It is not necessary to solve a dynamic program to obtain the optimal threshold values i_1 and i_2. Since we know the structural form is a control-limit policy, it is straightforward to derive the average cost for a given set of values of i_1 and i_2, and to solve for the optimal values of these thresholds by numerical methods. In particular, in the special case where $l_1 = r_1 = p$, (i.e., the "type" probabilities are the same for both sides), a closed-form solution can be obtained for the threshold values as stated in the following theorem:

Theorem 2 *Assume that $l_1 = r_1 = p$. Then the optimal threshold values are given by*

$$i_1 = -i_2 = \sqrt{2p(1-p)(V_{11} + V_{22} - V_{21} - V_{12})/h}. \tag{3.13}$$

Proof: The proof, as well as the cost resulting for a given set of thresholds is given in Appendix B.

4 Three and More Types

In this section, we characterize the optimal policy for the case where there are three types of left and right halves. The techniques used to prove the structure of the optimal policy for this case is then used to characterize the optimal policy structure for more than three types.

With three types of left and right halves, the state of the system can be represented by $n = \{n_1, n_2, n_3\}$, where n_t again equals the difference between the number of left and right-halves of type t. The vector n contains either one non-positive number and two non-negative numbers or one non-negative and two non-positive numbers. Letting $+$ denote a non-negative, and $-$ denote a non-positive number, there are six different situations of interest for n, namely, $\{+,+,-\}$, $\{+,-,+\}$, $\{-,+,+\}$, $\{-,-,+\}$, $\{-,+,-\}$, and $\{+,-,-\}$.

For brevity, we write the MDP formulation for the first situation (the formulation for the others is similar). Since $n_3 = -(n_1 + n_2)$, we need only represent the state in terms of n_1 and n_2. When $n_1 \geq 0$ and $n_2 \geq 0$,

$$g + f(n_1, n_2) = \min \begin{cases} P(n_1, n_2) + w(n_1, n_2) \\ P(n_1 - 1, n_2) - V_{13} + w(n_1 - 1, n_2) \\ P(n_1, n_2 - 1) - V_{23} + w(n_1, n_2 - 1) \end{cases} \tag{4.14}$$

and $g + f(0,0) = P(0,0) + w(0,0)$,

where

$$\begin{aligned} w(n_1, n_2) \equiv (l_1 r_1 + l_2 r_2 + l_3 r_3) f(n_1, n_2) + l_1 r_3 f(n_1 + 1, n_2) + l_2 r_3 f(n_1, n_2 + 1) \\ + l_1 r_2 f(n_1 + 1, n_2 - 1) + l_2 r_1 f(n_1 - 1, n_2 + 1) + l_3 r_1 f(n_1 - 1, n_2) + l_3 r_2 f(n_1, n_2 - 1) \end{aligned} \tag{4.15}$$

$$P(n_1, n_2) \equiv 2h|n_1 + n_2| + (\sum_{t=1}^{3} -l_t r_t V_{tt}) - l_1 r_2 V_{22} - l_2 r_1 V_{11}$$

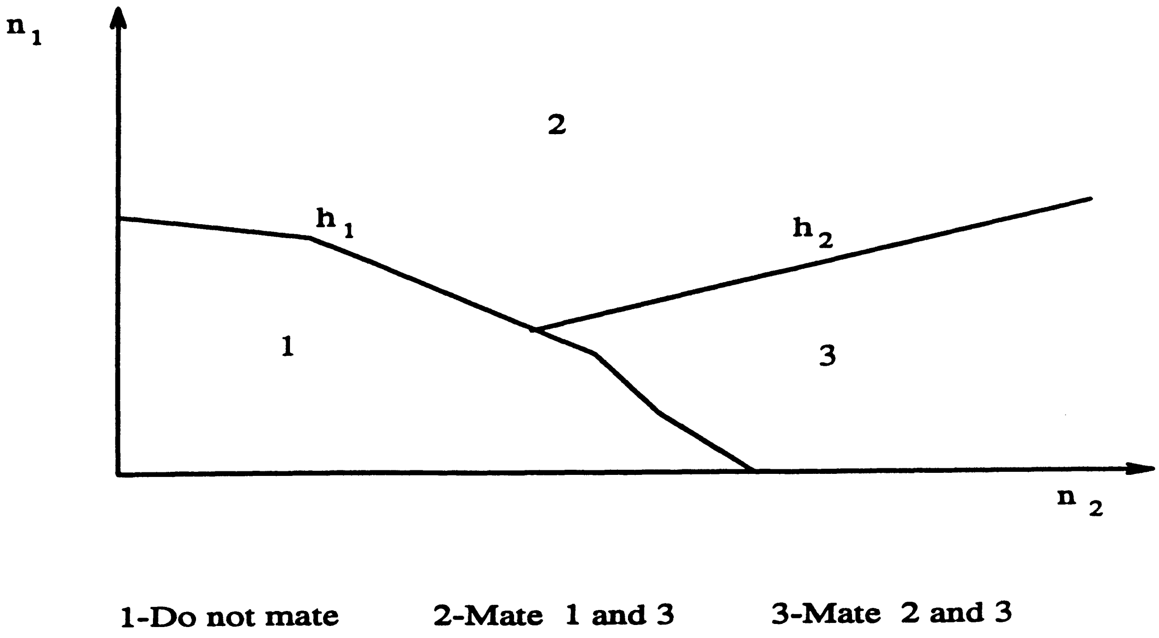

Figure 1: The structure of the optimal policy for three types

$$-l_3r_1(V_{11}+V_{33}) - l_3r_2(V_{22}+V_{33}) \quad \text{for } n_1 > 0,\ n_2 > 0 \tag{4.16}$$

$$P(0,n_2) \equiv 2h|n_2| + (\sum_{t=1}^{3} -l_tr_tV_{tt}) - l_1r_2V_{22} - l_3r_1V_{33} - l_3r_2(V_{22}+V_{33}) \tag{4.17}$$

$$P(n_1,0) \equiv 2h|n_1| + (\sum_{t=1}^{3} -l_tr_tV_{tt}) - l_2r_1V_{11} - l_3r_1(V_{11}+V_{33}) - l_3r_2V_{33} \tag{4.18}$$

$$P(0,0) \equiv \sum_{t=1}^{3} -l_tr_tV_{tt} \tag{4.19}$$

Equations (4.14) through (4.19) completely describe the MDP formulation for the situation where $n_1 \geq 0, n_2 \geq 0$.

We now describe the optimal policy for $n_1 \geq 0, n_2 \geq 0$. The *structure* of the optimal policy for the other 5 situations is the same. To see this, consider $n_1 \leq 0, n_2 \geq 0, n_3 \leq 0$. If we interchange the definition of types 2 and 3, and interchange the definition of left and right, we arrive at the situation where $n_1 \geq 0, n_2 \geq 0$. Thus, defining the *structure* of the optimal policy for only one of the six situations is sufficient to define the structure for all of them. (The details of the policies, however, are clearly not the same, since the values of V_{tu} might differ for each situation).

Theorem 3 below states that in each quadrant the optimal policy involves three regions separated by two monotonic functions:

Theorem 3 *The optimal policy divides the quadrant $n_1 \geq 0, n_2 \geq 0$ into three distinct regions defined by two functions $h_1(n_1)$ and $h_2(n_1)$. For any (n_1, n_2), if $n_2 < h_1(n_1)$ then the optimal policy is* not *to mate but to wait for the next right and left halves. If $n_2 \geq h_1(n_1)$ and $n_2 \geq h_2(n_1)$, then the policy is to mate a type 2 left-half with a type 3 right-half. Finally, if $n_2 \geq h_1(n_1)$ and $n_2 < h_2(n_1)$, then the policy is to mate a type 1 left half with a type 3 right-half. Furthermore, $h_1(n_1)$ is nonincreasing in n_1 and $h_2(n_1)$ is nondecreasing in n_1.*

Proof: The proof is in Appendix C.

Figure 1 is a sketch of the structure of the optimal policy. The two monotonic functions separate the plane into three inventory regions:

where it is optimal not to mate (Region 1); where it is optimal to mate a left half of type 1 with a right half of type 3 (Region 2); where it is optimal to mate a left half of type 2 with a right half of type 3 (Region 3). We note that in practice, the only "reachable" inventory positions will be either in Region 1, or on the surface of the curve separating this region from the other two regions. This is because mating will prevent inventory from ever reaching Regions 2 and 3.

Similar structural results can be obtained for more than 3 types. For example, consider 4 types, where all the left-sides are of types 1, 2 and 3 while the right-sides are of type 4. The optimal policy can be shown to be a 4-region policy where the 4-regions correspond to decisions: do nothing, mate 1 and 4, mate 2 and 4, and mate 3 and 4. Monotonicity properties can also be obtained for the functions that separate these four regions. As the number of types increases, the number of equations in the associated DP formulation grows exponentially. Therefore such structural results are of limited help in obtaining efficient solutions. In the next section, we focus on efficient heuristic solutions and test their performance by comparison to the optimal solution.

5 A Heuristic Solution

The MDP formulation of the dynamic type matching problem quickly suffers from the "curse-of-dimensionality" as the number of types increases. For example, when there are 5 types, formulating an MDP where the number of left or right sides of a certain type is at most 40 requires over 40,000,000 states. Therefore, for realistically sized problems, solving the MDP is not a practical solution. This leads us to consider two heuristic solutions.

The first heuristic was actually being used by our client plant at the time we became aware of the type mating problem. The policy was to wait until the sum of the left and right halves reach a predetermined inventory level, and then solve a transportation problem to decide how to mate *all* of the available halves. The optimal inventory level at which all halves were mated was determined by a simulation study. A small predetermined level leads to many unprofitable matings whereas a large level produces large holding costs. One disadvantage of this policy is that even if "matches" were available before the predetermined inventory level was reached, the policy did not allow matching those halves. A variation of this heuristic where halves of the same type are matched as soon as they are available is referred to as heuristic H1. The main drawback of H1 is that it requires simulation to decide *when* halves will be mated, and then solving a transportation problem every time the parts have to be mated.

A second heuristic, H2, assigns threshold values a_{tu} for mating each left side of type t with right side of type u. If both the number of left sides of type t and right sides of type u are greater than or equal to a_{tu}, then a left side of type t is mated with a right side of type u. This heuristic replaces the curves defining the regions of the optimal policy by simple rectangular regions. Figure 2 shows the decision regions for both the optimal policy and H2 for a 3-type problem. Note that for a 2-type problem, H2 is optimal.

To compute the thresholds a_{tu}, a 2-type mating problem is solved for all possible combinations, $t, u = 1, \ldots, T, t \neq u$. That is, the heuristic ignores all part types except t and u when computing a_{tu}. To do this, the production probabilities are first conditioned on

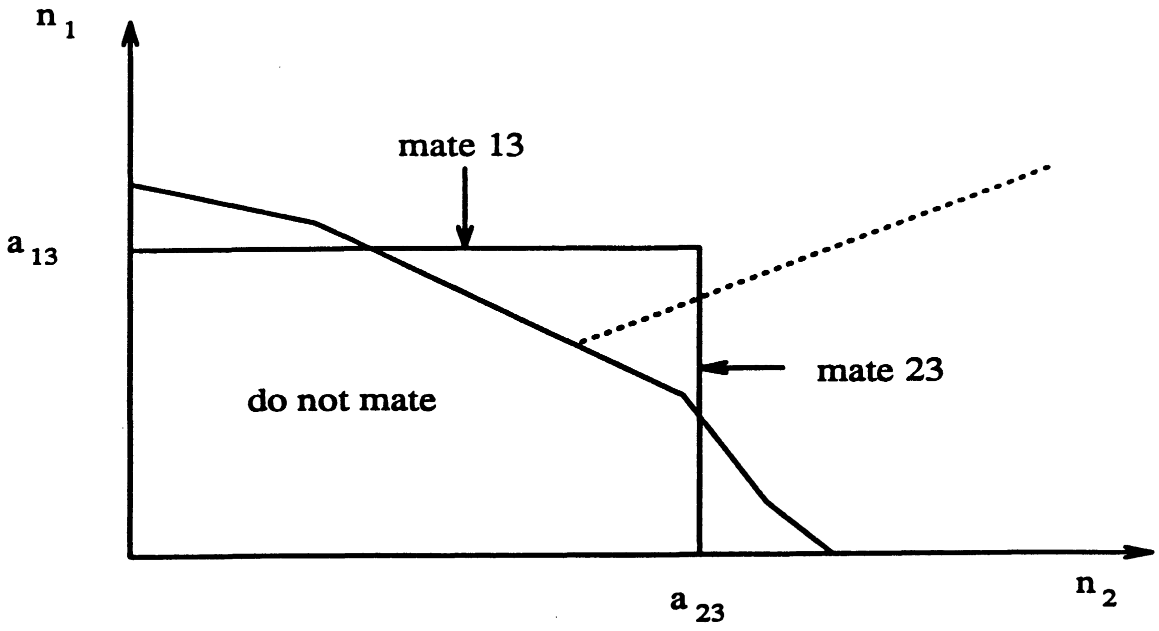

Figure 2: The structures of the optimal policy and heuristic H2 for three types

producing only t and u:

$$l_t^{'} \equiv \Pr\{\text{left type } t \text{ produced} \mid \text{only left } t \text{ or } u \text{ produced }\} = \frac{l_t}{l_t + l_u} \equiv 1 - l_u^{'} \quad (5.20)$$

$$r_t^{'} \equiv \Pr\{\text{ right type } t \text{ produced} \mid \text{only right } t \text{ or } u \text{ produced }\} = \frac{r_t}{r_t + r_u} \equiv 1 - r_u^{'} \quad (5.21)$$

The holding costs also must be re-scaled. In a 2-type mating problem, a new left and right half arrives every time unit, and h is the cost that each unit incurs until the arrival of another half of either type. With more than 2 types, the expected time until the arrival of another left half of type t or u equals $1/(l_t + l_u)$. Thus, each left half of type t or u in inventory incurs an expected holding cost of $h/(l_t + l_u)$ until the arrival of the next left half of either type. Similarly, each right side of type t or u incurs expected cost $h/(r_t + r_u)$ until the next arrival of either type. The rescaled holding cost is arbitrarily assigned to be the average of these values: $h' \equiv \frac{h}{2(l_t+l_u)} + \frac{h}{2(r_t+r_u)}$. Given the conditional probabilities, $l_t^{'}, r_t^{'}, l_u^{'}, r_u^{'}$, the rescaled holding cost h', and the original values $V_{tu}, V_{ut}, V_{tt}, V_{uu}$, a 2-type problem is solved to obtain the thresholds a_{tu} and a_{ut}.

The implementation of the heuristic is straightforward. Every time a new left half and right half is produced, left and right halves (if any) of the same type are matched. For any $t = 1, \ldots, T$, and $u = 1, \ldots, T$, if there are at least a_{tu} left halves of type t and right halves of type u, then a left side of type t is mated with a right side of type u.

H2 is at least as simple to implement in practice as H1, and the thresholds can be computed very quickly. For problems with as many as 16 types, the computation of *all* of the thresholds (which have to be computed only once) takes less than a minute on a 486 machine using a fairly unsophisticated code. Checking to see if the inventory of any type of part has exceeded the threshold is also very simple, especially in computerized electronics manufacturing environments.

6 Computational Results

A simulation study was conducted to test the performance of the two heuristics. Since problems with more than 4 types become extremely difficult to solve optimally, the two heuristics were tested against the optimal policy only for problems with 4 types. We also compared the two heuristics for typical 16-type problems arising in flat panel display manufacturing.

To test the performance of the heuristic, we chose three scenarios for the value functions, four scenarios for the left and right side probabilities, and three scenarios for holding costs. We tested both heuristics on all 36 combinations of scenarios.

Value Function Scenarios

Cases 1-12 represent situations where the value of the mated part gets lower as the "difference" between the halves increases (e.g., different types correspond to different tolerances, and the halves with the same tolerances have the best fit). Cases 13-24 represent situations where there are different qualities associated with each type, and an assembled component is only worth as much as its lowest quality part. Finally, Cases 25-36 represent a type mating problem where "passive" and "active" plates are cut into two after being mated. In this case, since each piece could have defects in two different locations, there are four possible types of plates. A type 1 plate has no defects, a type 4 plate has defects on both display locations. Similarly, a type 2 (type 3) plate is one that has a defect on its first (second) display location. Each good display is assumed to bring in a revenue of \$10, while

Case	$V_{11},...,V_{14}; V_{21},...,V_{24}$ $V_{31},...,V_{34}; V_{41},...,V_{44}$	l_1,l_2,l_3,l_4 r_1,r_2,r_3,r_4	h	H1 (% subopt.)	H2 (% subopt.)	Optimal
1	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.25,0.25,0.25,0.25; 0.25,0.25,0.25,0.25	0.02	9.47 (1.3)	9.57 (0.2)	9.59
2	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.25,0.25,0.25,0.25; 0.25,0.25,0.25,0.25	0.05	9.17 (2.0)	9.31 (0.5)	9.36
3	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.25,0.25,0.25,0.25; 0.25,0.25,0.25,0.25	0.10	8.83 (3.0)	9.05 (0.5)	9.10
4	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.30,0.30,0.30,0.10; 0.10,0.30,0.30,0.30	0.02	8.63 (1.5)	8.74 (0.2)	8.76
5	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.30,0.30,0.30,0.10; 0.10,0.30,0.30,0.30	0.05	8.46 (2.6)	8.66 (0.3)	8.69
6	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.30,0.30,0.30,0.10; 0.10,0.30,0.30,0.30	0.10	8.28 (3.6)	8.52 (0.8)	8.59
7	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.40,0.20,0.20,0.20; 0.40,0.20,0.20,0.20	0.02	9.47 (1.3)	9.57 (0.2)	9.59
8	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.40,0.20,0.20,0.20; 0.40,0.20,0.20,0.20	0.05	9.18 (1.9)	9.32 (0.4)	9.36
9	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.40,0.20,0.20,0.20; 0.40,0.20,0.20,0.20	0.10	8.84 (2.9)	9.06 (0.4)	9.10
10	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.22,0.32,0.12,0.34; 0.19,0.11,0.40,0.30	0.02	8.92 (3.0)	8.77 (4.7)	9.20
11	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.22,0.32,0.12,0.34; 0.19,0.11,0.40,0.30	0.05	8.69 (3.8)	8.69 (3.8)	9.03
12	10,8,6,4; 8,10,8,6; 6,8,10,8; 4,6,8,10	0.22,0.32,0.12,0.34; 0.19,0.11,0.40,0.30	0.10	8.43 (4.6)	8.56 (3.2)	8.84
13	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.25,0.25,0.25,0.25; 0.25,0.25,0.25,0.25	0.02	5.83 (1.7)	5.91 (0.3)	5.93
14	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.25,0.25,0.25,0.25; 0.25,0.25,0.25,0.25	0.05	5.59 (2.6)	5.72 (0.3)	5.74
15	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.25,0.25,0.25,0.25; 0.25,0.25,0.25,0.25	0.10	5.34 (3.6)	5.49 (0.9)	5.54
16	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.30,0.30,0.30,0.10; 0.10,0.30,0.30,0.30	0.02	5.35 (2.0)	5.45 (0.1)	5.46
17	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.30,0.30,0.30,0.10; 0.10,0.30,0.30,0.30	0.05	5.22 (3.2)	5.35 (0.7)	5.39
18	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.30,0.30,0.30,0.10; 0.10,0.30,0.30,0.30	0.10	5.07 (4.2)	5.25 (0.8)	5.29
19	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.40,0.20,0.20,0.20; 0.40,0.20,0.20,0.20	0.02	6.58 (1.5)	6.67 (0.1)	6.68
20	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.40,0.20,0.20,0.20; 0.40,0.20,0.20,0.20	0.05	6.35 (2.3)	6.48 (0.3)	6.50
21	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.40,0.20,0.20,0.20; 0.40,0.20,0.20,0.20	0.10	6.11 (3.0)	6.24 (1.0)	6.30
22	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.22,0.32,0.12,0.34; 0.19,0.11,0.40,0.30	0.02	5.02 (3.6)	4.94 (5.2)	5.21
23	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.22,0.32,0.12,0.34; 0.19,0.11,0.40,0.30	0.05	4.84 (4.7)	4.88 (3.9)	5.08
24	10,7.5,5,2.5; 7.5,7.5,5,2.5; 5,5,5,2.5; 2.5,2.5,2.5,2.5	0.22,0.32,0.12,0.34; 0.19,0.11,0.40,0.30	0.10	4.65 (5.7)	4.77 (3.2)	4.93

Table 1: Results for 4-type Cases

Case	$V_{11},...,V_{14}; V_{21},...,V_{24}$	l_1,l_2,l_3,l_4	h	H1	H2	Optimal
	$V_{31},...,V_{34}; V_{41},...,V_{44}$	r_1,r_2,r_3,r_4		(% subopt.)	(% subopt.)	
25	20,12,12,4; 12,12,4,4;	0.25,0.25,0.25,0.25;				
	12,4,12,4; 4,4,4,4	0.25,0.25,0.25,0.25	0.02	11.35 (1.0)	11.46 (0.0)	11.47
26	20,12,12,4; 12,12,4,4;	0.25,0.25,0.25,0.25;				
	12,4,12,4; 4,4,4,4	0.25,0.25,0.25,0.25	0.05	10.95 (2.0)	11.13 (0.3)	11.17
27	20,12,12,4; 12,12,4,4;	0.25,0.25,0.25,0.25;				
	12,4,12,4; 4,4,4,4	0.25,0.25,0.25,0.25	0.10	10.52 (2.9)	10.78 (0.5)	10.83
28	20,12,12,4; 12,12,4,4;	0.30,0.30,0.30,0.10;				
	12,4,12,4; 4,4,4,4	0.10,0.30,0.30,0.30	0.02	10.16 (1.8)	10.35 (0.0)	10.35
29	20,12,12,4; 12,12,4,4;	0.30,0.30,0.30,0.10;				
	12,4,12,4; 4,4,4,4	0.10,0.30,0.30,0.30	0.05	9.94 (3.2)	10.25 (0.2)	10.27
30	20,12,12,4; 12,12,4,4;	0.30,0.30,0.30,0.10;				
	12,4,12,4; 4,4,4,4	0.10,0.30,0.30,0.30	0.10	9.68 (4.4)	10.10 (0.3)	10.13
31	20,12,12,4; 12,12,4,4;	0.40,0.20,0.20,0.20;				
	12,4,12,4; 4,4,4,4	0.40,0.20,0.20,0.20	0.02	12.96 (1.0)	13.06 (0.2)	13.09
32	20,12,12,4; 12,12,4,4;	0.40,0.20,0.20,0.20;				
	12,4,12,4; 4,4,4,4	0.40,0.20,0.20,0.20	0.05	12.57 (1.7)	12.76 (0.2)	12.79
33	20,12,12,4; 12,12,4,4;	0.40,0.20,0.20,0.20;				
	12,4,12,4; 4,4,4,4	0.40,0.20,0.20,0.20	0.10	12.16 (2.4)	12.39 (0.6)	12.46
34	20,12,12,4; 12,12,4,4;	0.22,0.32,0.12,0.34;				
	12,4,12,4; 4,4,4,4	0.19,0.11,0.40,0.30	0.02	8.93 (1.7)	9.08 (0.0)	9.08
35	20,12,12,4; 12,12,4,4;	0.22,0.32,0.12,0.34;				
	12,4,12,4; 4,4,4,4	0.19,0.11,0.40,0.30	0.05	8.73 (3.1)	9.01 (0.0)	9.01
36	20,12,12,4; 12,12,4,4;	0.22,0.32,0.12,0.34;				
	12,4,12,4; 4,4,4,4	0.19,0.11,0.40,0.30	0.10	8.48 (4.8)	8.91 (0.0)	8.91

Table 2: Results for 4-type Cases

each defective one has a salvage value of $2.

Type Probability Scenarios

We consider four different scenarios for the type probabilities. They range from completely balanced situations where the probability for each type is equal to 0.25 for the left and right sides, to more unbalanced cases where the left and right side probabilities are not equal for any of the types.

Holding Cost Scenarios

We also consider three different holding costs ranging from a low of 0.02 to a high of 0.10.

Heuristic Performance for 36 Combinations of Scenarios

Tables 1 and 2 show that both heuristics perform well for the 36 cases. The average suboptimality of H2 was under 1 % over the 36 cases, while the average suboptimality of H1 was 2.8 %. The fact that H2 works so well is encouraging since it is much easier to compute the parameters of H2: the thresholds required to implement H2 were calculated in several seconds. By comparison, the simulation runs needed for H1 (required to compute the inventory level at which parts are mated) took as long as 15 minutes on a 486 machine.

Interestingly, the V for which H2 performed best was for Cases 25-36 (flat panel display problems with two displays per plate). The reason for this is the very sharp drop in value associated with mating plates that include defects in either location. This sharp drop in value creates an incentive to keep high inventories of active (passive) plates of a given type in the hope that the same type of passive (active) plate will eventually be produced. The thresholds for mating are therefore high, and a small error in estimating these high threshold has little effect on performance. In Cases 1-24, both H1 and H2 performed worse on average when the left and right side probabilities were different than when they were the same. These preliminary results indicate that the performance of the heuristics depends on both the value function and the probability vector used (with better performance when the probabilities are balanced, and the thresholds for mating are not very low). The holding cost does not have an appreciable effect on performance of the heuristic.

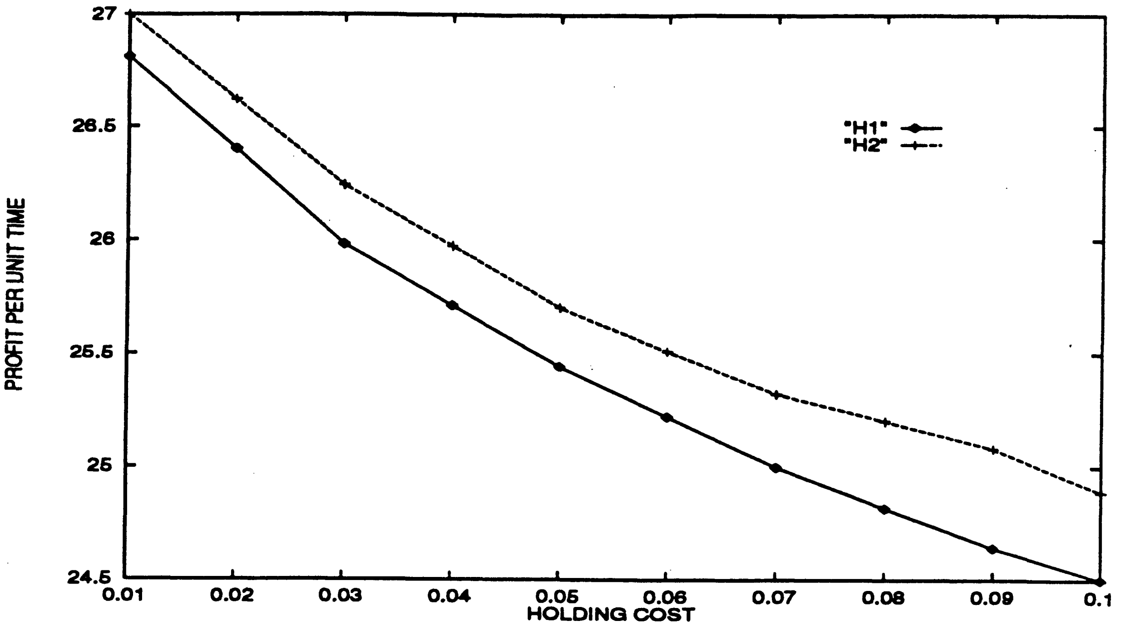

Figure 3: Performance of H1 and H2 for 16-type problem

Heuristic Performance for a Practical Example

Finally, the two heuristics were compared on a typical problem from flat panel display manufacturing, (The example given here is very similar to ones observed in practice, although the actual data was changed for reasons of confidentiality). We use N=4 displays, which results in T=16 possible types of halves. The probability of defect in any location is 0.3, independent of defects at other locations. A display is worth $10 if it has no defects on either of its halves. Therefore mating defect-free "passive" and "active" halves is worth $40. On the other hand, mating an active plate that has a defect in the location of the third display with a passive plate that has defects in the location of the second and fourth displays will result in production of only one nondefective display (in the location of the first display) and the total mate is worth only $10.

Since this problem has 16 types, it cannot be solved to optimality using dynamic programming. Even if the maximum inventory of a given type were limited to be at most 40, there would be over 10^{28} states in the MDP. Figure 3 shows the performance of H2 and H1 as a function of h. For all of the holding cost values displayed, H2 outperforms H1. Before implementing a variation of H1 our client plant had considered mating plates as they became available, and not holding any in inventory. For this example problem this policy would result in an average profit of \$19.6 per unit time, which is roughly 30 % below the profit that H2 achieves, and 25 % below the profit achieved by H1. The actual improvement observed by our client plant when they used H1 was similar.

7 Conclusions and Further Research

In this paper, we formulated a dynamic type mating problem that arises in many manufacturing environments including flat panel display manufacturing. We provided structural results and presented two effective heuristics for this previously unaddressed problem. The first heuristic (H1) has already been implemented in practice. However, the second heuristic (H2) is simpler to implement and initial results indicate that it is potentially more effective than H1. Further research is needed to test the effectiveness of these and other heuristics on larger scale problems.

Further research is needed to derive structural results and heuristics for more general value functions than those considered here. When matching parts of the same type immediately is no longer optimal, the problem becomes much more difficult, and does not seem to have an easily identifiable structure. It is possible to implement a version of H1 in this case which would wait until N halves are produced, and once again use the transportation algorithm to decide how parts will be mated. In this case, matching would not be allowed (until all N parts are produced). The performance of this and other possible heuristics needs to be investigated.

Further research should also address the case where there is uncertainty with respect to the *time* of arrival as well as the type of the next subassembly. This is a more difficult problem than that addressed here and the structure of the optimal policy seems to be very complicated even in the case of only two types.

Appendix A:

To prove Theorem 1, we first present the following lemmas:

Lemma 1 *If the following three conditions hold:*

1a) $w(i+1) - 2w(i) + w(i-1) \geq 0$ *for all* $i > 1, i < -1$

1b) $w(2) - 2w(1) + w(0) \geq -l_2 r_1(V_{11} + V_{22})$

1c) $w(-2) - 2w(-1) + w(0) \geq -l_1 r_2(V_{11} + V_{22})$,

the optimal policy for the 2-type matching problem is the "control-limit" policy:

a) mate a left half of type 1 with a right-half of type 2 if $i > i_1 \geq 0$,

b) mate a left-half of type 2 with a right-half of type 1 if $i < i_2 \leq 0$, *and*

c) wait for the next produced halves if $i_2 < i < i_1$.

Proof: Suppose that, for some finite $i > 1$, the optimal decision is to mate a left-half of type 1 and a right-half of type 2. Note that such a finite i will exist, since otherwise $f_i = \infty$ $\forall i$, which contradicts optimality of f_i compared to any arbitrary finite-i matching policy.

Then, by (3.8),

$$w(i) \geq -V_{12} - 2h + w(i-1) \qquad (A1)$$

Rewriting condition (1a) as $w(i+1) - w(i) \geq w(i) - w(i-1)$, and adding this to (A1) gives $w(i+1) \geq -V_{12} - 2h + w(i)$. By (3.8), this implies that the decision is to mate in state $i+1$. The proof for the case where $i < -1$ is analogous.

Suppose that the decision is to mate in state $i = 1$. Then, by (3.10), $w(1) \geq -V_{12} - 2h + l_2 r_1(V_{11} + V_{22}) + w(0)$. Writing condition (1b) as $w(2) - w(1) \geq w(1) - w(0) - l_2 r_1(V_{11} + V_{22})$, and adding these two inequalities results in $w(2) \geq -V_{12} - 2h + w(1)$. By (3.8), the optimal decision is to mate also in state 2. The argument for states $i = -1$ and $i = 0$ is the same.

□

Lemma 2 *If and only if the* $w(i)$ *satisfy (1a), (1b), and (1c), the following relations hold*

2a) $f_{i+1} - 2f_i + f_{i-1} \geq 0$ *for all* $i > 0$ *and* $i < 0$

2b) $f_1 - 2f_0 + f_{-1} \geq -V_{11} - V_{22}$.

Proof: We first prove that inequalities (2a) and (2b) imply (1a)-(1c)..

Consider $i > 1$ or $i < -1$. If condition (2a) holds, then condition (1a) follows immediately from (3.5). To show (1b) when (2a) and (2b) hold, we write, using (3.5)

$$\begin{aligned} w(2) - 2w(1) + w(0) &= (l_1 r_1 + l_2 r_2)(f_2 - 2f_1 + f_0) + l_1 r_2 (f_3 - 2f_2 + f_1) + \\ &\quad l_2 r_1 (f_1 - 2f_0 + f_{-1}) \\ &\geq -l_2 r_1 (V_{11} + V_{22}) \end{aligned}$$

The proof that (1c) holds is similar.

To prove that the inequalities (1a-1c) imply (2a-2b), first note that we can rewrite (3.8) as

$$g + f_i = P^+(i) - V_{12} - 2h + w(i-1) + \min\{w(i) - w(i-1) + V_{12} + 2h; 0\}$$

This implies that

$$\begin{aligned} f_{i+1} - f_i &= P^+(i+1) - P^+(i) + w(i) - w(i-1) + \min\{w(i+1) - w(i) + V_{12} + 2h; 0\} \\ &\quad - \min\{w(i) - w(i-1) + V_{12} + 2h; 0\}. \end{aligned}$$

For $i > 1$, $P^+(i+1) - P^+(i) = 2h$, and we can rewrite the above expression as

$$f_{i+1} - f_i = 2h + \min\{w(i+1) - w(i) + V_{12} + 2h, 0\} + \max\{-V_{12} - 2h; w(i) - w(i-1)\}$$

By assumption (1a), $w(i) - w(i-1)$ is increasing for all $i > 1$ which proves that $f_{i+1} - 2f_i + f_{i-1} \geq 0$ for all $i > 1$. The proofs for the other cases are similar. □.

Theorem 1 is now proved by using those lemmas.

Proof of Theorem 1: The proof is by convergence. Consider an iterative procedure for solving equations (3.8) through (3.12) rewritten for $k > 0$ as

$$g + f_i^k = P^+(i) + \min\{w^k(i); -V_{12} - 2h + w^k(i-1)\} \quad \text{for } i > 1 \qquad (A2)$$

$$g + f_i^k = P^-(i) + \min\{w^k(i); -V_{21} - 2h + w^k(i+1)\} \quad \text{for } i < -1 \qquad (A3)$$

$$g + f_1^k = P^+(1) + \min\{w^k(1); -V_{12} - 2h + w^k(0) + l_2 r_1(V_{11} + V_{22}) \qquad (A4)$$

$$g + f_{-1}^k = P^-(-1) + \min\{w^k(-1); -V_{21} - 2h + w^k(0) + l_1 r_2(V_{11} + V_{22}) \qquad (A5)$$

$$g + f_0^k = P^+(0) + w^k(0) + l_2 r_1(V_{11} + V_{22}) = P^-(0) + w^k(0) + l_1 r_2(V_{11} + V_{22}) \qquad (A6)$$

In addition, we write (3.5) as

$$w^k(i) \equiv l_1 r_1 f_i^{k-1} + l_2 r_2 f_i^{k-1} + l_1 r_2 f_{i+1}^{k-1} + l_2 r_1 f_{i-1}^{k-1} \qquad (A7)$$

Begin by setting $w^0(i) = 0$ for all i. In general, f_i^k, can be computed from $w^k(i)$ through the use of equations (A2) through (A6), and $w^k(i)$ can in turn be computed from f_i^{k-1} by using (A7). Since $w^0(i) = 0$ satisfies the condition of Lemma 1, we are guaranteed that at each iteration the f and w values will satisfy the conditions of Lemmas 1 and 2. By Theorem 2.2 of Hernandez-Lerma (1989), $w^k(i) \to w(i)$, and $f_i^k \to f_i$. The proof therefore follows from convergence and Lemma 1. □.

Appendix B:

In this appendix, we compute the average return of the policy that mates halves whenever there are i_1 units of type 1 left and type 2 right sides or whenever there are i_2 units of type 2 left and type 1 right sides. For ease of notation, let $x = i_1$ and $y = -i_2$

Using these thresholds x and y, the system can be represented by a Markov chain with $x + y - 1$ states $\{-y+1, \ldots, 0, \ldots, x-1\}$. A transition from any state j to $j+1$ occurs if a left type 1 and a right type 2 is produced, an event that has probability $a \equiv l_1 r_2$. A transition from state j to $j-1$ has by a similar argument, probability $b \equiv l_2 r_1$. With probability $l_1 r_1 + l_2 r_2$, the next arriving parts will be of the same type and the system remains in state j.

It is straightforward to compute π_k, the long-run probability that this chain is in state k:

$$\pi_k = \frac{1-(a/b)}{1-(a/b)^{x+y-1}}(a/b)^{k+y-1} \quad \text{if } a \neq b \tag{B1}$$

$$\pi_k = 1/(x+y-1) \quad \text{if } a = b \tag{B2}$$

Since in state k there are $|2k|$ units of inventory on hand, the average inventory in the system, $I(x,y)$ is:

$$I(x,y) = \sum_{k=1-y}^{x-1} |2k|\pi_k. \tag{B3}$$

To compute the expected revenue per period, we note that in each state revenue of V_{11} will be earned with probability l_1r_1, and a revenue of V_{22} will be earned with probability l_2r_2. In addition, in states $k > 0$, (i.e., when there are k type 1 left sides and type 2 right sides) $V_{11} + V_{22}$ will also be earned when the next arrivals are a left side of type 2 and right side of type 1 (which occurs with probability $b = l_1r_2$). Similarly, in states $k < 0$, $V_{11} + V_{22}$ will be earned with probability a. Finally, in state $x - 1(-y+1)$, mating will be required with probability $a(b)$ resulting in a reward of $V_{12}(V_{21})$. Therefore, the total expected revenue per period is:

$$V(x,y) = l_1r_1V_{11}+l_2r_2V_{22}+(V_{11}+V_{22})(a\sum_{k=-y+1}^{-1}\pi_k+b\sum_{k=1}^{x-1}\pi_k)+aV_{12}\pi_{x-1}+bV_{21}\pi_{-y+1}. \tag{B4}$$

The total expected return per unit time including inventory cost can then be expressed as

$$g(x,y) = V(x,y) - hI(x,y) \tag{B5}$$

Numerical methods can be used to compute x and y.

In the special case when $l_1 = r_1 = p$, the Markov chain is doubly stochastic, so that $\pi_k = 1/(x+y-1)$ for all k. Equation (B3) becomes

$$I(x,y) = \frac{x(x-1)+y(y-1)}{x+y-1}$$

and (B4) is

$$V(x,y) = p^2V_{11} + (1-p)^2V_{22} + p(1-p)\frac{(V_{11}+V_{22})(x+y-2)+V_{21}+V_{12}}{x+y-1}$$

Since both $I(x,y)$ and $V(x,y)$ are completely symmetric with respect to x and y, at the optimal profit-maximizing value we can set $x = y$. Therefore, the total return as a function of x is

$$g(x) = p^2V_{11} + (1-p)^2V_{22} + p(1-p)\frac{(V_{11}+V_{22}(2x-1)+V_{21}+V_{12})}{2x-1} - 2h\frac{x(x-1)}{2x-1}$$

Since g is concave in x, the maximizing value can be found by differentiating with respect to x, setting equal to 0, and taking the appropriate closest integer, the result of which is given in (3.13).

Appendix C:

To prove Theorem 3, we first prove the following technical lemmas:

Lemma 3 *The optimal policy for the 3-type matching problem is a three region policy for each quadrant if the following conditions for the case* $(n_1 \geq 0, n_2 \geq 0)$, *and analogous conditions for the other 5 cases, hold:*

1a) $w(n_1+1,n_2) - w(n_1,n_2) \geq w(n_1,n_2) - w(n_1-1,n_2)$ *for* $n_1 \geq 2$.

1b) $w(2,n_2) - w(1,n_2) \geq w(1,n_2) - w(0,n_2) - l_2r_1V_{11} - l_3r_1V_{11}$ *for* $n_2 > 0$

1c) $w(2,0) - w(1,0) \geq w(1,0) - w(0,0) - r_1(l_2+l_3)V_{11} - l_3(r_1+r_2)V_{33}$

2a) $w(n_1,n_2+1) - w(n_1,n_2) \geq w(n_1,n_2) - w(n_1,n_2-1)$ *for* $n_2 \geq 2$

2b) $w(n_1,2) - w(n_1,1) \geq w(n_1,1) - w(n_1,0) - l_1r_2V_{22} - l_3r_2V_{22}$ *for* $n_1 > 0$

2c) $w(0,2) - w(0,1) \geq w(0,1) - w(0,0) - r_2(l_1+l_3)V_{22} - l_3(r_1+r_2)V_{33}$

3a) $w(n_1+1,n_2-1) - w(n_1,n_2-1) \geq w(n_1,n_2) - w(n_1-1,n_2)$ *for* $n_1 > 1, n_2 \geq 1$

3b) $w(2,n_2-1) - w(1,n_2-1) \geq w(1,n_2) - w(0,n_2) - l_2r_1V_{11} - l_3r_1V_{11}$ *for* $n_2 \geq 1$.

4a) $w(n_1 - 1, n_2 + 1) - w(n_1 - 1, n_2) \geq w(n_1, n_2) - w(n_1, n_2 - 1) \quad \textit{for } n_1 \geq 1, n_2 > 1$

4b) $w(n_1 - 1, 2) - w(n_1 - 1, 1) \geq w(n_1, 1) - w(n_1, 0) - l_1 r_2 V_{22} - l_3 r_2 V_{22} \quad \textit{for } n_1 \geq 1$

5a) $w(n_1, n_2 + 1) - w(n_1, n_2) \geq w(n_1 - 1, n_2 + 1) - w(n_1 - 1, n_2) \ \textit{for } n_1 > 1, n_2 \geq 1$

5b) $w(1,1) - w(1,0) \geq w(0,1) - w(0,0) - l_3 r_1 V_{33} - l_3 r_2 V_{33}$

Proof: The three feasible decisions available when $n_1 \geq 0, n_2 \geq 0$ are a) mating a left half of type 1 with a right half of type 3, denoted (M13); b) mating a left half of type 2 with a right-half of type 3 (M23); and c) not-mating this period (NM). Conditions (1a-1c) ensure that if M13 is preferable to NM in state (n_1, n_2), then it is also preferable in state $(n_1 + 1, n_2)$. Similarly, conditions (2a-2c) ensure that if M23 is preferable to NM in state (n_1, n_2), then it is also preferable in state $(n_1, n_2 + 1)$. Conditions (3a) and (3b) guarantee that if M13 is preferable to M23 in state (n_1, n_2), it is also preferable in state $(n_1 + 1, n_2)$, while conditions (4a) and (4b) ensure that if M23 is preferable to M13 in state (n_1, n_2), it is also preferable in state $(n_1, n_2 + 1)$. Finally, conditions (5a) and (5b) ensure that if NM is not optimal in state (n_1, n_2) that it also can not be optimal in either $(n_1 + 1, n_2)$ or $(n_1, n_2 + 1)$.

By an argument similar to that in the proof of Lemma 1, conditions (3a-3b) and (4a-4b) are sufficient to guarantee the existence and monotonicity of $h_2(.)$, while conditions (1a-1c), (2a-2c) and (5a-5b) are sufficient to guarantee the existence and monotonicity of $h_1(.)$. □.

Lemma 4 *Conditions (1a-5b) of Lemma 3, and the equivalent conditions for the other 5 cases, hold if and only if the following conditions hold:*

1a) $f(n_1 + 1, n_2) - f(n_1, n_2) \geq f(n_1, n_2) - f(n_1 - 1, n_2) \quad \textit{for } n_1 \geq 1,$

1b) $f(1, n_2 - f(0, n_2) \geq f(0, n_2) - f(-1, n_2) - V_{11} \quad \textit{for } n_2 \geq 0$

1c) $f(1,0) - f(0,0) \geq f(0,0) - f(-1,0) - V_{11} - V_{33}$

2a) $f(n_1, n_2 + 1) - f(n_1, n_2) \geq f(n_1, n_2) - f(n_1, n_2 - 1) \quad \textit{for } n_2 \geq 1$

2b) $f(n_1,1) - f(n_1,0) \geq f(n_1,0) - f(n_1,-1) - V_{22} \quad for\ n_1 \geq 0$

2c) $f(0,1) - f(0,0) \geq f(0,0) - f(0,-1) - V_{22} - V_{33}$

3a) $f(n_1+1,n_2-1) - f(n_1,n_2-1) \geq f(n_1,n_2) - f(n_1-1),n_2) \quad for\ n_1 > 0, n_2 \geq 1$

3b) $f(1,n_2-1) - f(0,n_2-1) \geq f(0,n_2) - f(-1,n_2) - V_{11} \quad for\ n_2 \geq 1$

4a) $f(n_1-1,n_2+1) - f(n_1-1,n_2) \geq f(n_1,n_2) - f(n_1,n_2-1) \quad for\ n_1 \geq 1, n_2 > 0$

4b) $f(n_1-1,1) - f(n_1-1,0) \geq f(n_1,0) - f(n_1,-1) - V_{22}\ for\ n_1 \geq 1$

5a) $f(n_1,n_2+1) - f(n_1,n_2) \geq f(n_1-1,n_2+1) - f(n_1-1,n_2)\ for\ n_1 \geq 1, n_2 \geq 0$

5b) $f(0,1) - f(0,0) \geq f(-1,1) - f(-1,0) - V_{33}$

5c) $f(1,0) - f(1,-1) \geq f(0,0) - f(0,-1) - V_{33}$

Proof: That if conditions (1a-5c) for f imply the conditions (1a-5c) for w in Lemma 3 is easily verified by using (4.15).

To prove that the conditions for w imply the conditions for f, we need to prove each case individually. For example, to prove that condition (1a) for w implies condition (1a) for f, we rewrite (4.14) as

$$\begin{aligned} g + f(n_1,n_2) &= P(n_1,n_2) + \min\{w(n_1,n_2); -2h - V_{13} + w(n_1-1,n_2); \\ &\quad -2h - V_{23} + w(n_1,n_2-1)\} \end{aligned} \tag{C1}$$

since for $n_1 \geq 2, n_2 \geq 2$, $P(n_1,n_2) - P(n_1-1,n_2) = 2h$. We can rewrite (C1) as

$$\begin{aligned} g + f(n_1,n_2) &= P(n_1,n_2) + \min\{-2h - V_{13} + w(n_1-1,n_2); -2h - V_{23} + w(n_1,n_2-1) \\ &\quad + \min\{0, w(n_1,n_2) + 2h + V_{23} - w(n_1,n_2-1)\}\} \end{aligned} \tag{C2}$$

$$g + f(n_1,n_2) = P(n_1,n_2) - 2h - V_{13} + w(n_1-1,n_2) + \varphi(n_1,n_2) \tag{C3}$$

where

$$\begin{aligned} \varphi(n_1, n_2) \;\; &\equiv \;\; \min\{0; -V_{23} + V_{13} + w(n_1, n_2 - 1) - w(n_1 - 1, n_2) \\ &\quad + \min\{0; w(n_1, n_2) + 2h + V_{23} - w(n_1, n_2 - 1)\}\}. \end{aligned} \qquad (C4)$$

Therefore, using (C3), we can write

$$f(n_1, n_2) - f(n_1 - 1, n_2) = w(n_1 - 1, n_2) - w(n_1 - 2, n_2) + \varphi(n_1, n_2) - \varphi(n_1 - 1, n_2) \quad (7.22)$$

The term $\varphi(n_1, n_2)$ is increasing in n_1 since by assumption (5a) for w, $w(n_1, n_2) - w(n_1, n_2 - 1)$ is increasing in n_1, and by assumption (3a), $w(n_1, n_2 - 1) - w(n_1 - 1, n_2)$ is increasing in n_1, and the minimum of increasing functions is increasing. Therefore, we only need to show that

$$Z(n_1, n_2) = w(n_1 - 1, n_2) - w(n_1 - 2, n_2) - \varphi(n_1 - 1, n_2)$$

is increasing in n_1. We can rewrite $Z(n_1, n_2)$ as

$$\begin{aligned} Z(n_1, n_2) \;\; &= \;\; \max\{w(n_1 - 1, n_2) - w(n_1 - 2, n_2); \max\{V_{23} - V_{13} + w(n_1 - 1, n_2) - w(n_1 - 1, n_2 - 1); \\ &\quad -2h - V_{13}\}\}. \end{aligned}$$

The function $Z(n_1, n_2)$ is increasing in n_1 since $w(n_1 - 1, n_2) - w(n_1 - 2, n_2)$ is increasing in n_1 by assumption (1a) for w, and $w(n_1 - 1, n_2) - w(n_1 - 1, n_2 - 1)$ is increasing by assumption (5a) for w. Therefore $f(n_1, n_2) - f(n_1 - 1, n_2)$ is increasing in n_1, and we have shown that conditions (1a-5b) for w imply condition (1a) for f. The proof for the other cases is similar. □.

We can then use Lemmas 3 and 4 in a recursive manner similar to that in the proof of Theorem 1 to obtain a convergence argument that proves Theorem 3.

Acknowledgements:

We are grateful to Professor Hau Lee and two anonymous referees for their comments which greatly improved the paper. We would also like to thank Professor Yavuz Bozer for his many suggestions. This work was supported in part by Grant No: DDM-9308290 from the National Science Foundation and a grant from the Center for Display Technology and Manufacturing at the University of Michigan.

Bibliography

[1] Ammar, M.H., 1980. "Modelling and Analysis of Unreliable Manufacturing Assembly Networks with Finite Storage," MIT Laboratory for Information and Decision Sciences, Report, LIDS-TH-1004.

[2] Bhat, U.N., 1986. "Finite Capacity Assembly-like Queues," *Queueing Systems, Theory and Applications,* **1**, 85.

[3] Bonomi, F. 1987. "An Approximate Analysis for a Class of Assembly-like Queues," *Queueing Systems: Theory and Applications,* **1**, 289.

[4] Duenyas, I, 1994. "Estimating the Throughput of a Cyclic Assembly System," *International Journal of Production Research* **32**, 1403.

[5] Duenyas, I. and W.J. Hopp, 1992. "CONWIP Assembly with Deterministic Processing and Random Outages," *IIE Transactions,* **24**, 97.

[6] Duenyas, I. and W.J. Hopp, 1993. "Estimating the Throughput of an Exponential CONWIP Assembly System," *Queueing Systems: Theory and Applications* **14**, 135.

[7] Duenyas, I. and M.F. Keblis, 1995, "Release Policies for Assembly Systems," *IIE Transactions,* **27**, 507.

[8] Gershwin, S.B., 1991, "Assembly/disassembly systems: an efficient decomposition algorithm for tree structured networks," *IIE Transactions,* **23**, 302.

[9] Hopp, W.J., and J.T. Simon, 1989. "Bounds and Heuristics for Assembly-Like Queues," *Queueing Systems: Theory and Applications*, **4**, 137.

[10] Hernandez-Lerma, O. 1989. *Adaptive Markov Control Processes*, Springer-Verlag, New York.

[11] Iyama, T., Masahiro, M., Goto, S. and Koga, T., 1992. "A Race Matching Method for Ball Bearing Manufacture", Department of Mechanical Engineering, Iwate University, 4-3-5, Ueda, Morioka 020 Japan.

[12] Lee, H.S., and S.M. Pollock, 1989. "Approximate analysis for the merge configuration of an open queueing network with blocking," *IIE Transactions,* **21**, 122.

[13] Lipper, E.H., and B. Sengupta, 1986. "Assembly-like queues with finite capacity: bounds, asymptotics and approximations," *Queueing Systems: Theory and Applications* **1**, 67.

[14] Saboo, S., and W.E. Wilhelm, 1986. "An approach for modeling small-lot assembly networks," *IIE Transactions,* **18**, 322.